

METHOD FOR JUST-IN-TIME UPDATING OF PROGRAMMING PARTS

RELATED APPLICATIONS

[0001] This application is related to copending and cofiled applications for United States Letters Patent Serial No. _____, filed _____ and entitled METHOD FOR ACCESSING SCAN CHAINS AND UPDATING EEPROM-RESIDENT FPGA CODE THROUGH A SYSTEM MANAGEMENT PROCESSOR AND JTAG BUS (Attorney Docket No. 10017840-1); Serial No. _____ filed _____ and entitled METHOD AND APPARATUS FOR SERIAL BUS TO JTAG BUS BRIDGE (Attorney Docket No. 10017841-1); and Serial No. _____, filed _____ and entitled SYSTEM AND METHOD FOR IN-SYSTEM PROGRAMMING THROUGH AN ON-SYSTEM JTAG BRIDGE OF PROGRAMMABLE LOGIC DEVICES ON MULTIPLE CIRCUIT BOARDS OF A SYSTEM (Attorney Docket No. 10016250-1), all of the aforementioned applications incorporated herewith by reference thereto.

BACKGROUND OF THE INVENTION

[0002] Many systems and products incorporate programmable parts, such as FPGAs, PLDs, EEPROMS and microprocessors. Often, the programming for these parts changes during the development stage, prototyping stage and even into the production stage. It is important and yet difficult to incorporate the latest programming into the parts at each stage, particularly in the advanced manufacturing stages when several entities may be involved.

[0003] In the manufacturing stage, for example, an outside contract vendor may program these parts for systems and products designed by the originating company; a contract manufacturer may perform the actual integrations of the parts within the systems and products, for sale under label of the originating company. The difficulties occur when the originating company identifies technical issues with firmware that must be changed and incorporated into the integration and/or manufacturing processes. The firmware changes must be communicated first to the contract vendor, and then physically integrated as a change into the parts integrations.

[0004] Implementing programming changes from the originating company and the contract vendor and manufacturer is thus tedious and difficult; and yet any delay in programming parts within these products and systems can cause critical program problems for the originating company. Likewise, an incorrect programming of parts with firmware that is not the latest revision can also cause critical program problems; systems and products may become inoperable in such instances.

[0005] There are two general methods for updating programming to programmable parts within such systems and products, and according to the prior art:

1. New program revisions are sent from the originating company to the contracting vendor. The vendor programs a few parts and sends them back to the originating company for verification. Once verified, the contract vendor is authorized to program the parts in larger quantities; the contract vendor then ships parts to the contract manufacturer, where programmed parts with older revisions are purged and the newer revision parts are integrated within the processes.

This method has several drawbacks. First, changes according to this method can take several weeks. Second, there is often considerable confusion about which revision should be implemented into the systems or products. Third, each company – the originating company, the contract vendor, and the contract manufacturer – typically expends significant overhead and resources in implementing and verifying the method. And fourth, manufacturing costs significantly increase when large quantities of programmed parts are purged to prepare for new revisions of these parts.

2. Parts are programmed in In-Circuit Test (ICT). An ICT generally consists of a “bed of nails” configuration to access signals on a circuit board. An ICT can be used to test boards, or components, but it may also be used to program parts. For example, certain pin configurations to a programmable part may be used to program the part; while other pin configurations to the programmable part may be used to test the part. There are several difficulties with the ICT: First, a long development time usually results from constructing an ICT. Second, the typical ICT cannot operate at high speeds, lengthening the time to program the part. Third, the use of an ICT is not generally scaleable to program many parts; the ICT then may become a bottleneck to efficient manufacturing and inventories.

[0006] It is, accordingly, one object of the invention to provide methods for programming of parts through a network. Other objects of the invention are apparent within the description that follows.

SUMMARY OF THE INVENTION

[0007] In one aspect, the invention provides a method of releasing new firmware updates to a programmable part within a circuit board. The method includes the steps of creating an image file of firmware used to program the part; storing the image file at a server; integrating the programmable part with the printed circuit board; and networking the circuit board to the server such that the image file downloads to the circuit board for programming the programmable part. The method may include the further step of automatically polling the server to download the firmware to the circuit board.

[0008] In one aspect, the method includes the step of integrating a serial chip with the printed circuit board, the serial chip polling the server to download the firmware, the programmable part having bootstrap software to download the firmware from the serial chip to the programmable part.

[0009] In one aspect, the method includes the step of utilizing the Internet in networking the circuit board with the server. A LAN or WAN may also be used in networking the circuit board with the server, alone or in combination with the Internet, to accomplish the steps of the invention.

[0010] In another aspect, the method includes the further step of updating the firmware image file at the server, subsequent downloads of the firmware image file to a programmable part being seamless to the updated firmware.

[0011] In another aspect, a system is provided for programming programmable parts in a manufacturing line. A firmware server connects to a network; the server storing a firmware image file. An interface server with the manufacturing line connects to the network and captures the image file from the firmware server. A first system with a first programmable part interfaces with the interface server to program the first programmable part with the firmware image file. A second system with a second programmable part then interfaces with the interface server to program the second programmable part with the firmware image file. Alternatively, each of the systems has an interface server for integrating with the firmware server; the interface servers each downloading firmware to programmable parts associated with its system. Preferably, the interface server includes a connector for physically coupling with a circuit board within the first and second systems, the first and second programmable parts being electronically and respectively coupled with the first and second printed circuit boards. The connector has one or more pins that interface in a programming configuration with pads or pins on the printed circuit boards to program the

programmable parts. In the system, the image file may be updated or replaced with another image file without affecting interaction between the interface server and the system.

[0012] In still another aspect, the interface server has a connector with a plurality of pins for programming the part. The pins connect to the part, or to a connector of a printed circuit board housing the part, or to a system and signal lines coupled to the part, to program the part. The connector and pins may for example act similar to an ICT device.

[0013] In an alternative aspect, a system is provided for programming a series of programmable parts in a manufacturing line. A firmware server connects to a network; the server storing a firmware image file. An interface server connects to the network and captures the image file from the firmware server. Sequentially, one programmable part and then another interfaces with the interface server; the interface server programs each programmable part, in sequence, with the firmware image file. Preferably, the interface server includes a connector for physically coupling with pins of the programmable parts. The connector has one or more pins that interface in a programming configuration with the parts to facilitate programming the programmable parts.

[0014] In other aspects of the invention, multiple firmware image files are stored on a firmware server. Multiple interface servers connect in network with the firmware server to download one or more of the files to programmable parts associated with the interface servers. Typically, each of the interface servers resides within a product that includes at least one printed circuit board having the programmable parts; each of the servers downloading firmware for the programmable parts of that circuit board.

[0015] The invention is next described further in connection with preferred embodiments, and it will become apparent that various additions, subtractions, and modifications can be made by those skilled in the art without departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] A more complete understanding of the invention may be obtained by reference to the drawings, in which:

[0017] FIG. 1 shows a system constructed according to the invention for programming one or more programmable parts, in accord with the invention; and

[0018] FIG. 2 shows another system of the invention for programming programmable parts in accord with the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0019] FIG. 1 shows a system 10 constructed according to the invention. System 10 includes a firmware server 12 connected by a network 14 to an interface server 16. Network 14 may for example be the Internet, a LAN or WAN, or combinations thereof. Firmware server 12 stores a firmware image file 17; firmware image file 17 is used to program a programmable part 20 for use within a system 22. Communications between server 12 across network 14 and to interface server 16 may be secure link enabled by a virtual private network or other private link.

[0020] System 22 may for example be a product incorporating a printed circuit board 24 (PCB 24). Programmable part 20 may and preferably is integrated with printed circuit board 24. Programmable part 20 may for example be a FPGA, PLD, EEPROM or other programmable logic; programmable part 20 may alternatively be a programmable processor like a PIC processor from MICRO CHIP, a 68HC05 family processor from MOTOROLA, a 68HC11 processor from MOTOROLA, and a 8051 or 8096 processors from INTEL.

[0021] Interface server 16 interfaces with programmable part 20 to program part 20 according to image file 17. In the preferred embodiment, interface server 16 has a connector 30 to physically connect with a connector 25 of PCB 24, to program part 20. Accordingly, connector 30 has a plurality of pins 32 arranged in a programming configuration to couple with PCB 24.

[0022] Those skilled in the art should appreciate that system 10 may program part 20 with alternative connections to connector 30. For example, connector 30 may connect directly to part 20, in a pin-specific configuration of pins 32, to program part 20. Alternatively, connector 30 may connect with system 22 – and specifically by connection of pins 32 to internal signal lines 34 connected to part 20 – to equivalently program part 20 in the appropriate programming configuration.

[0023] It should therefore be clear that system 10 is illustrative to show one or more possible configurations for programming part 20. Specifically, in one configuration, interface server 16 connects directly to PCB 24 by connector pins 32 of connector 30 to program part 20 within PCB 24. In a second configuration, interface server 16 couples to system 22 by connector pins 32 coupled to signal lines 34; signal lines 34 thus connect in the right programming configuration to PCB 24. In the third configuration, interface server 16 couples directly to part 20 via its connector and pins 30, 32, respectively. In this third

configuration, therefore, PCB 24 and system 22 are not part of the process and are not, thus, included during the step of programming part 20.

[0024] System 10 preferably operates within a manufacturing process with many parts 20 programmed in sequence (or to some extent concurrently, if multiple servers 16 are within the manufacturing process). FIG. 1 thus illustratively shows system 10 with a robotics link 60 that shuttles sequential parts 20 for programming by interface server 16. Depending on the configuration, link 60 may sequentially shuttle parts 20 for programming, or PCBs 24 with parts 20 for programming, or systems 22 with PCBs and parts 24, 20, for programming. FIG. 1 shows a second system 22', second PCB 24' and part 20' illustrating a second of the many parts 20 programmed in sequence.

[0025] In one embodiment, PCB 24 includes a serial or memory device 26 used to poll and/or capture firmware image file 17 from server 16 automatically.

[0026] Firmware image file 17 is updated by overwriting a similarly named image file to firmware server 12. An engineer desiring to update image file 17 to a newer revision for programming part 20 can thus update part 20 by communicating the later revision firmware from his computer 70 connected to firmware server 12. Updating the firmware 17 in this manner is "seamless" to the programming process at the location of interface server 16; that is, continued programming of parts 20 may occur without special configuration or communication indicating the newer revision.

[0027] Interface server 16 may be, and preferably is, integral with system 22; in addition, the invention may program more than a single part within a PCB. FIG. 2 illustrates this preferred embodiment in a system 100 of the invention. System 100 includes a firmware server 112 connected by a network 114 to one or more interface servers 116. System 100 is shown to connect to two interface servers 116A, 116B simultaneously; system 100 may further connect to additional or fewer servers 116, as a matter of design choice.

[0028] Network 114 may for example be the Internet, a LAN or WAN, or combinations thereof. Secure communications between servers 116 and firmware server 112 may occur through one of known techniques, as a matter of design choice. Firmware server 112 stores a series of firmware image files 17A1 ... 17AN, 117B1 ... 117BN; firmware image files 117 are used to program a corresponding number of programmable parts 20A1...120AN, 120B1 ... 120BN. The number of files 117 and corresponding parts 120 is a matter of design choice; those skilled in the art should appreciate that system 100 may for example program one or more parts 120 without departing from the scope of the invention.

[0029] Systems 122 may for example be a product incorporating a printed circuit board 124. System 100 is shown to include two systems 122A, 122B, each with a corresponding interface server 116A, 116B and boards 124A, 124B; however system 100 may further include additional or fewer systems 122 as a matter of design choice.

[0030] Programmable parts 120 may and preferably are integrated with printed circuit boards 124. Programmable parts 120 may for example be a FPGA, PLD, EEPROM or other programmable logic; programmable parts 120 may alternatively be a programmable processor.

[0031] Interface servers 116 interface with programmable parts 120 to program parts 120 according to image files 117. In the preferred embodiment, interface server 116 resides within system 122 and connects to program parts 120 via one or more signal lines 121 (for example, server 116A connects to parts 120A within system 122A; server 116B connects to parts 120B within system 122B).

[0032] System 100 preferably operates within a manufacturing process with many parts 120 programmed concurrently. Firmware image files 117 may be updated by overwriting a similarly named image file to firmware server 112. An engineer desiring to update one or more of image files 117 to a newer revision for one or more of programming parts 120 may thus update the parts by communicating the later revision firmware from his computer 170 connected to firmware server 112. Updating firmware 117 in this manner is "seamless" to the programming process at the location of interface server 116; that is, continued programming of parts 120 may occur without special configuration or communication indicating the newer revision.

[0033] Those skilled in the art should appreciate that interface servers 116 may embody built-in parts and/or components within systems 122, or within boards 124, to accomplish the same function, without departing from the scope of the invention. Specifically, interface servers 116 may be, or may include, smart components or processors coupled with printed circuit boards 124; such smart components or processors then directly communicate with the network firmware server 112.

[0034] The inventions of FIG. 1 and FIG. 2 have several advantages. First, systems 10, 100 are very fast as compared to prior art methods for updating firmware for programmable parts, such as those using the ICT. Second, a single firmware image is preferably used throughout the manufacturing life cycle of system 22, 122; this eliminates the overhead and tracking of revisions for various firmware upgrades to parts. Third, the process

of updating firmware to programmable parts is less disruptive to the manufacturing process, increasing product throughput and reducing error rates.

[0035] The invention thus attains the objects set forth above, among those apparent from the preceding description. Since certain changes may be made in the above methods and systems without departing from the scope of the invention, it is intended that all matter contained in the above description or shown in the accompanying drawing be interpreted as illustrative and not in a limiting sense. It is also to be understood that the following claims are to cover all generic and specific features of the invention described herein, and all statements of the scope of the invention which, as a matter of language, might be said to fall there between.